CSCI567 Machine Learning (Fall 2018)

Prof. Haipeng Luo

U of Southern California

Sep 26, 2018

Administration

HW 2 released, due on 10/07

HW 1 solutions are available, W1 is graded, P1 grading should be done by the end of this week

Administration

HW 2 released, due on 10/07

HW 1 solutions are available, W1 is graded, P1 grading should be done by the end of this week

Midterm

- 10/03, 5PM-7PM, *THH 101 and THH 201* (pick any one), sit every other seat, closed-book/notes, no electronics allowed
- Multiple choice + regular problems like homework
- Coverage: mostly Lec 1-5, some simple questions from Lec 6 (today)

Outline

1. Review of last lecture

2. Support vector machines (primal formulation)

3. A detour of Lagrangian duality

4. Support vector machines (dual formulation)

Outline

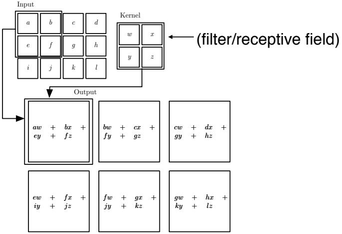
1. Review of last lecture
2. Support vector machines (primal formulation)
3. A detour of Lagrangian duality
4. Support vector machines (dual formulation)

Convolutional Neural Nets

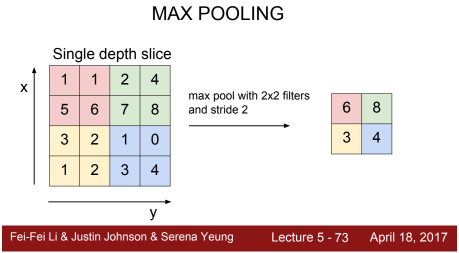
Typical architecture for CNNs:

Input → [[Conv → ReLU]*N → Pool?]*M → [FC → ReLU]*Q → FC

2D Convolution



(Goodfellow 2016)



Kernel functions

Definition: a function $k : \mathbb{R}^{\mathsf{D}} \times \mathbb{R}^{\mathsf{D}} \rightarrow \mathbb{R}$ is called a *(positive semidefinite) kernel function* if there exists a function $\boldsymbol{\phi} : \mathbb{R}^{\mathsf{D}} \rightarrow \mathbb{R}^{\mathsf{M}}$ so that for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{\mathsf{D}}$,

$$k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}')$$

Kernel functions

Definition: a function $k : \mathbb{R}^{\mathsf{D}} \times \mathbb{R}^{\mathsf{D}} \rightarrow \mathbb{R}$ is called a *(positive semidefinite) kernel function* if there exists a function $\boldsymbol{\phi} : \mathbb{R}^{\mathsf{D}} \rightarrow \mathbb{R}^{\mathsf{M}}$ so that for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{\mathsf{D}}$,

$$k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}')$$

Examples we have seen

$$k(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}')^2$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{d=1}^{\mathsf{D}} \frac{\sin(2\pi(x_d - x'_d))}{x_d - x'_d}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}' + c)^d \qquad \text{(polynomial kernel)}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = e^{-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2}{2\sigma^2}} \qquad \text{(Gaussian/RBF kernel)}$$

Kernelizing ML algorithms

Feasible as long as **only inner products are required**:

- regularized linear regression (dual formulation)

$$\phi(x)^{\mathrm{T}} w^* = \phi(x)^{\mathrm{T}} \Phi^{\mathrm{T}} (K + \lambda I)^{-1} y \quad (K = \Phi\Phi^{\mathrm{T}} \text{ is } \textit{kernel matrix})$$

- nearest neighbor classifier with L2 distance

$$\|\phi(x) - \phi(x')\|_2^2 = k(x, x) + k(x', x') - 2k(x, x')$$

- perceptron, logistic regression, SVM, ...

Outline

Support vector machines (SVM)

- One of the most commonly used classification algorithms
- Works well with the kernel trick
- Strong theoretical guarantees

Support vector machines (SVM)

- One of the most commonly used classification algorithms
- Works well with the kernel trick
- Strong theoretical guarantees

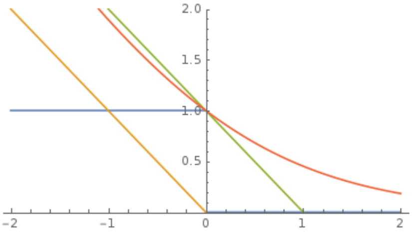
We focus on **binary classification** here.

Primal formulation

In one sentence: linear model with L2 regularized hinge loss.

Primal formulation

In one sentence: linear model with L2 regularized hinge loss. Recall



- perceptron loss $\ell_{\text{perceptron}}(z) = \max\{0, -z\} \rightarrow$ Perceptron
- logistic loss $\ell_{\text{logistic}}(z) = \log(1 + \exp(-z)) \rightarrow$ logistic regression
- hinge loss $\ell_{\text{hinge}}(z) = \max\{0, 1 - z\} \rightarrow$ **SVM**

Primal formulation

For a linear model $(\boldsymbol{w}, b)$, this means

$$\min_{\boldsymbol{w},b} \sum_n \max\left\{0, 1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b)\right\} + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

- recall $y_n \in \{-1, +1\}$
- a nonlinear mapping $\boldsymbol{\phi}$ is applied
- the bias/intercept term b is used explicitly (think about why after this lecture)

Primal formulation

For a linear model $(\boldsymbol{w}, b)$, this means

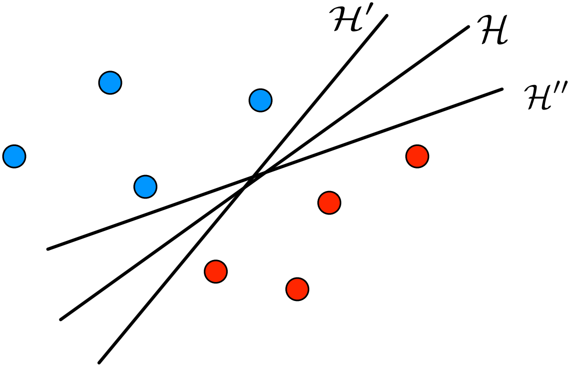
$$\min_{\boldsymbol{w},b} \sum_n \max\left\{0, 1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b)\right\} + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

- recall $y_n \in \{-1, +1\}$
- a nonlinear mapping ϕ is applied
- the bias/intercept term b is used explicitly (think about why after this lecture)

So why L2 regularized hinge loss?

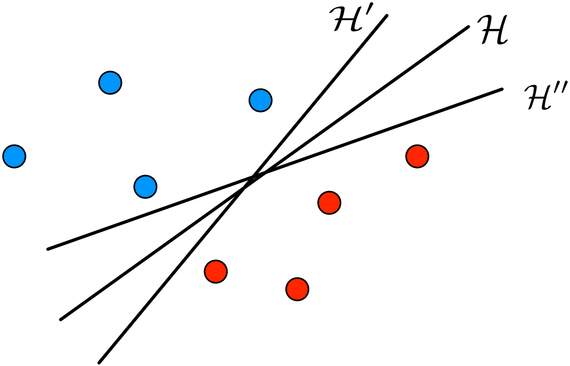
Geometric motivation: separable case

When data is **linearly separable**, there are *infinitely many hyperplanes with zero training error*:



Geometric motivation: separable case

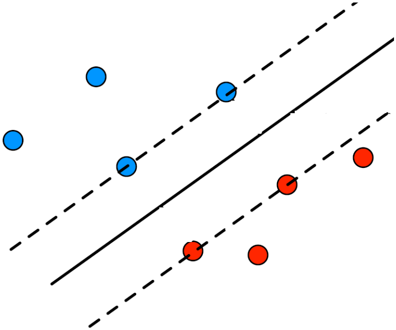
When data is **linearly separable**, there are *infinitely many hyperplanes with zero training error*:



So which one should we choose?

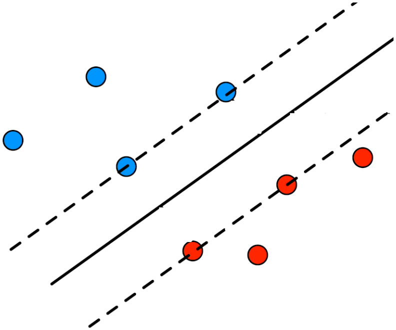
Intuition

The further away from data points the better.



Intuition

The further away from data points the better.



How to formalize this intuition?

Distance to hyperplane

What is the **distance** from a point $\boldsymbol{x}$ to a hyperplane $\{\boldsymbol{x} : \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b = 0\}$?

Distance to hyperplane

What is the **distance** from a point $\boldsymbol{x}$ to a hyperplane $\{\boldsymbol{x} : \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b = 0\}$?

Assume the **projection** is $\boldsymbol{x} - \ell\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}$, then

$$0 = \boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{x} - \ell\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}\right) + b = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} - \ell\|\boldsymbol{w}\| + b$$

and thus $\ell = \frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}{\|\boldsymbol{w}\|_2}$.

Distance to hyperplane

What is the **distance** from a point $\boldsymbol{x}$ to a hyperplane $\{\boldsymbol{x} : \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b = 0\}$?

Assume the **projection** is $\boldsymbol{x} - \ell\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}$, then

$$0 = \boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{x} - \ell\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}\right) + b = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} - \ell\|\boldsymbol{w}\| + b$$

and thus $\ell = \frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}{\|\boldsymbol{w}\|_2}$.

Therefore the distance is

$$\frac{|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b|}{\|\boldsymbol{w}\|_2}$$

Distance to hyperplane

What is the **distance** from a point $\boldsymbol{x}$ to a hyperplane $\{\boldsymbol{x} : \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b = 0\}$?

Assume the **projection** is $\boldsymbol{x} - \ell\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}$, then

$$0 = \boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{x} - \ell\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}\right) + b = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} - \ell\|\boldsymbol{w}\| + b$$

and thus $\ell = \frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}{\|\boldsymbol{w}\|_2}$.

Therefore the distance is

$$\frac{|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b|}{\|\boldsymbol{w}\|_2}$$

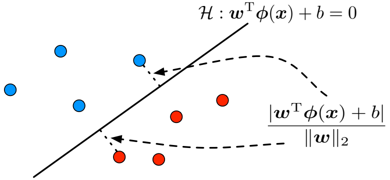
For a hyperplane that correctly classifies $(\boldsymbol{x}, y)$, the distance becomes

$$\frac{y(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b)}{\|\boldsymbol{w}\|_2}$$

Maximizing margin

Margin: the *smallest* distance from all training points to the hyperplane

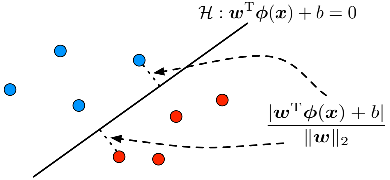
$$\text{MARGIN OF } (\boldsymbol{w}, b) = \min_n \frac{y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b)}{\|\boldsymbol{w}\|_2}$$



Maximizing margin

Margin: the *smallest* distance from all training points to the hyperplane

$$\text{MARGIN OF } (\boldsymbol{w}, b) = \min_n \frac{y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b)}{\|\boldsymbol{w}\|_2}$$



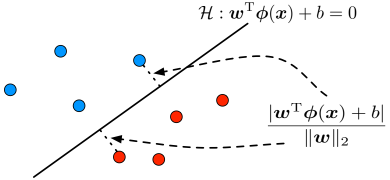
The intuition "**the further away the better**" translates to solving

$$\max_{\boldsymbol{w}, b} \quad \min_n \frac{y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b)}{\|\boldsymbol{w}\|}$$

Maximizing margin

Margin: the *smallest* distance from all training points to the hyperplane

$$\text{MARGIN OF } (\boldsymbol{w}, b) = \min_n \frac{y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b)}{\|\boldsymbol{w}\|_2}$$



The intuition "**the further away the better**" translates to solving

$$\max_{\boldsymbol{w},b} \quad \min_n \frac{y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b)}{\|\boldsymbol{w}\|} = \max_{\boldsymbol{w},b} \frac{1}{\|\boldsymbol{w}\|_2} \min_n y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b)$$

Rescaling

Note: rescaling $(\boldsymbol{w}, b)$ does not change the hyperplane at all.

Rescaling

Note: rescaling $(\boldsymbol{w}, b)$ does not change the hyperplane at all.

We can thus always scale $(\boldsymbol{w}, b)$ s.t. $\min_n y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) = 1$

Rescaling

Note: rescaling $(\boldsymbol{w}, b)$ does not change the hyperplane at all.

We can thus always scale $(\boldsymbol{w}, b)$ s.t. $\min_n y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) = 1$

The margin then becomes

$$
\begin{aligned}
&\textsc{margin of } (\boldsymbol{w},\, b) \\
&= \frac{1}{\|\boldsymbol{w}\|_2} \min_n y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \\
&= \frac{1}{\|\boldsymbol{w}\|_2}
\end{aligned}
$$

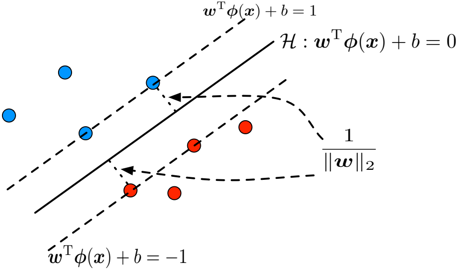
Rescaling

Note: rescaling $(\boldsymbol{w}, b)$ does not change the hyperplane at all.

We can thus always scale $(\boldsymbol{w}, b)$ s.t. $\min_n y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) = 1$

The margin then becomes

$$\begin{aligned}
&\text{MARGIN OF } (\boldsymbol{w},\ b) \\
&= \frac{1}{\|\boldsymbol{w}\|_2} \min_n y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \\
&= \frac{1}{\|\boldsymbol{w}\|_2}
\end{aligned}$$



Summary for separable data

For a separable training set, we aim to solve

$$\max_{\boldsymbol{w},b} \frac{1}{\|\boldsymbol{w}\|_2} \quad \text{s.t.} \quad \min_n y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) = 1$$

Summary for separable data

For a separable training set, we aim to solve

$$\max_{\boldsymbol{w},b} \frac{1}{\|\boldsymbol{w}\|_2} \quad \text{s.t.} \quad \min_n y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) = 1$$

This is equivalent to

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \geq 1, \quad \forall\, n \end{aligned}$$

Summary for separable data

For a separable training set, we aim to solve

$$\max_{\boldsymbol{w},b} \frac{1}{\|\boldsymbol{w}\|_2} \quad \text{s.t.} \quad \min_n y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) = 1$$

This is equivalent to

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \geq 1, \quad \forall\, n \end{aligned}$$

SVM is thus also called *max-margin* classifier. The constraints above are called *hard-margin* constraints.

General non-separable case

If data is not linearly separable, the previous constraint

$$y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \geq 1, \quad \forall\ n$$

is obviously *not feasible*.

General non-separable case

If data is not linearly separable, the previous constraint

$$y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \geq 1, \quad \forall\ n$$

is obviously *not feasible*.

To deal with this issue, we relax them to **soft-margin** constraints:

$$y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \geq 1 - \xi_n, \quad \forall\ n$$

where we introduce **slack variables** $\xi_n \geq 0$.

SVM Primal formulation

We want ξ_n to be as small as possible too.

SVM Primal formulation

We want ξ_n to be as small as possible too. The objective becomes

$$\begin{aligned} \min_{\boldsymbol{w}, b, \{\xi_n\}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C \sum_n \xi_n \\ \text{s.t.} \quad & y_n(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b) \geq 1 - \xi_n, \quad \forall\, n \\ & \xi_n \geq 0, \quad \forall\, n \end{aligned}$$

where C is a hyperparameter to balance the two goals.

Equivalent form

Formulation

$$
\begin{aligned}
\min_{\boldsymbol{w}, b, \{\xi_n\}} \quad & C \sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\
\text{s.t.} \quad & 1 - y_n(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b) \leq \xi_n, \quad \forall\ n \\
& \xi_n \geq 0, \quad \forall\ n
\end{aligned}
$$

Equivalent form

Formulation

$$\begin{aligned} \min_{\boldsymbol{w}, b, \{\xi_n\}} \quad & C \sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & 1 - y_n(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b) \leq \xi_n, \quad \forall\ n \\ & \xi_n \geq 0, \quad \forall\ n \end{aligned}$$

is equivalent to

$$\begin{aligned} \min_{\boldsymbol{w}, b, \{\xi_n\}} \quad & C \sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & \max\left\{0, 1 - y_n(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b)\right\} = \xi_n, \quad \forall\ n \end{aligned}$$

Equivalent form

$$
\begin{aligned}
\min_{\boldsymbol{w}, b, \{\xi_n\}} \quad & C \sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\
\text{s.t.} \quad & \max\left\{0, 1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b)\right\} = \xi_n, \quad \forall\, n
\end{aligned}
$$

is equivalent to

$$
\min_{\boldsymbol{w}, b} \; C \sum_n \max\left\{0, 1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b)\right\} + \frac{1}{2}\|\boldsymbol{w}\|_2^2
$$

Equivalent form

$$\begin{aligned} \min_{\boldsymbol{w},b,\{\xi_n\}} \quad & C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & \max\left\{0, 1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n)+b)\right\} = \xi_n, \quad \forall\ n \end{aligned}$$

is equivalent to

$$\min_{\boldsymbol{w},b}\ C\sum_n \max\left\{0, 1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n)+b)\right\} + \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

and

$$\min_{\boldsymbol{w},b}\ \sum_n \max\left\{0, 1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n)+b)\right\} + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

with $\lambda = 1/C$.

Equivalent form

$$\min_{\boldsymbol{w}, b, \{\xi_n\}} \quad C \sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2$$
$$\text{s.t.} \quad \max\left\{0, 1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b)\right\} = \xi_n, \quad \forall\, n$$

is equivalent to

$$\min_{\boldsymbol{w}, b} \ C \sum_n \max\left\{0, 1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b)\right\} + \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

and

$$\min_{\boldsymbol{w}, b} \ \sum_n \max\left\{0, 1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b)\right\} + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

with $\lambda = 1/C$. *This is exactly minimizing L2 regularized hinge loss!*

Optimization

$$\begin{aligned} \min_{\boldsymbol{w}, b, \{\xi_n\}} \quad & C \sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & 1 - y_n(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b) \leq \xi_n, \quad \forall\ n \\ & \xi_n \geq 0, \quad \forall\ n \end{aligned}$$

- It is a convex (**quadratic** in fact) problem

Optimization

$$\min_{\boldsymbol{w}, b, \{\xi_n\}} \quad C \sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

$$\text{s.t.} \quad 1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \leq \xi_n, \quad \forall\ n$$

$$\xi_n \geq 0, \quad \forall\ n$$

- It is a convex (**quadratic** in fact) problem
- thus can apply any convex optimization algorithms, e.g. SGD

Optimization

$$\begin{aligned} \min_{\boldsymbol{w}, b, \{\xi_n\}} \quad & C \sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & 1 - y_n(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b) \leq \xi_n, \quad \forall\ n \\ & \xi_n \geq 0, \quad \forall\ n \end{aligned}$$

- It is a convex (**quadratic** in fact) problem
- thus can apply any convex optimization algorithms, e.g. SGD
- there are **more specialized and efficient** algorithms

Optimization

$$\begin{aligned} \min_{\boldsymbol{w},b,\{\xi_n\}} \quad & C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & 1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \le \xi_n, \quad \forall\ n \\ & \xi_n \ge 0, \quad \forall\ n \end{aligned}$$

- It is a convex (**quadratic** in fact) problem
- thus can apply any convex optimization algorithms, e.g. SGD
- there are **more specialized and efficient** algorithms
- but usually we apply kernel trick, which requires solving the *dual problem*

Outline

Lagrangian duality

Extremely important and powerful tool in analyzing optimizations

Lagrangian duality

Extremely important and powerful tool in analyzing optimizations

We will introduce basic concepts and derive the **KKT conditions**

Lagrangian duality

Extremely important and powerful tool in analyzing optimizations

We will introduce basic concepts and derive the **KKT conditions**

Applying it to SVM reveals an important aspect of the algorithm

Primal problem

Suppose we want to solve

$$\min_{\boldsymbol{w}} F(\boldsymbol{w}) \quad \text{s.t. } h_j(\boldsymbol{w}) \leq 0 \quad \forall\, j \in [\mathsf{J}]$$

where functions $h_1, \ldots, h_{\mathsf{J}}$ define J **constraints**.

Primal problem

Suppose we want to solve

$$\min_{\boldsymbol{w}} F(\boldsymbol{w}) \quad \text{s.t. } h_j(\boldsymbol{w}) \le 0 \quad \forall\, j \in [\mathsf{J}]$$

where functions $h_1, \ldots, h_{\mathsf{J}}$ define J **constraints**.

SVM primal formulation is clearly of this form with $\mathsf{J} = 2\mathsf{N}$ constraints:

$$F(\boldsymbol{w}, b, \{\xi_n\}) = C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

$$h_n(\boldsymbol{w}, b, \{\xi_n\}) = 1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) - \xi_n \quad \forall\, n \in [\mathsf{N}]$$

$$h_{\mathsf{N}+n}(\boldsymbol{w}, b, \{\xi_n\}) = -\xi_n \quad \forall\, n \in [\mathsf{N}]$$

Lagrangian

The **Lagrangian** of the previous problem is defined as:

$$L(\boldsymbol{w}, \{\lambda_j\}) = F(\boldsymbol{w}) + \sum_{j=1}^{\mathsf{J}} \lambda_j h_j(\boldsymbol{w})$$

where $\lambda_1, \ldots, \lambda_{\mathsf{J}} \geq 0$ are called **Lagrangian multipliers**.

Lagrangian

The **Lagrangian** of the previous problem is defined as:

$$L\left(\boldsymbol{w}, \{\lambda_j\}\right) = F(\boldsymbol{w}) + \sum_{j=1}^{\mathsf{J}} \lambda_j h_j(\boldsymbol{w})$$

where $\lambda_1, \ldots, \lambda_{\mathsf{J}} \geq 0$ are called **Lagrangian multipliers**.

Note that

$$\max_{\{\lambda_j\} \geq 0} L(\boldsymbol{w}, \{\lambda_j\}) = \begin{cases} F(\boldsymbol{w}) & \text{if } h_j(\boldsymbol{w}) \leq 0 \quad \forall\, j \in [\mathsf{J}] \\ +\infty & \text{else} \end{cases}$$

Lagrangian

The **Lagrangian** of the previous problem is defined as:

$$L\left(\boldsymbol{w}, \{\lambda_j\}\right) = F(\boldsymbol{w}) + \sum_{j=1}^{\mathsf{J}} \lambda_j h_j(\boldsymbol{w})$$

where $\lambda_1, \ldots, \lambda_{\mathsf{J}} \geq 0$ are called **Lagrangian multipliers**.

Note that

$$\max_{\{\lambda_j\}\geq 0} L(\boldsymbol{w}, \{\lambda_j\}) = \begin{cases} F(\boldsymbol{w}) & \text{if } h_j(\boldsymbol{w}) \leq 0 \quad \forall\, j \in [\mathsf{J}] \\ +\infty & \text{else} \end{cases}$$

and thus,

$$\min_{\boldsymbol{w}} \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}, \{\lambda_j\}\right) \iff \min_{\boldsymbol{w}} F(\boldsymbol{w}) \ \text{ s.t. } \ h_j(\boldsymbol{w}) \leq 0 \quad \forall\, j \in [\mathsf{J}]$$

Duality

We define the **dual problem** by swapping the min and max:

$$\max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

Duality

We define the **dual problem** by swapping the min and max:

$$\max_{\{\lambda_j\} \geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

How are the primal and dual connected?

Duality

We define the **dual problem** by swapping the min and max:

$$\max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

How are the primal and dual connected? Let $\boldsymbol{w}^*$ and $\{\lambda\}_j^*$ be the primal and dual solutions respectively

Duality

We define the **dual problem** by swapping the min and max:

$$\max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

How are the primal and dual connected? Let $\boldsymbol{w}^*$ and $\{\lambda\}_j^*$ be the primal and dual solutions respectively, then

$$\max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right) = \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j^*\}\right)$$

Duality

We define the **dual problem** by swapping the min and max:

$$\max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

How are the primal and dual connected? Let $\boldsymbol{w}^*$ and $\{\lambda\}_j^*$ be the primal and dual solutions respectively, then

$$\max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right) = \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j^*\}\right) \leq L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right)$$

Duality

We define the **dual problem** by swapping the min and max:

$$\max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

How are the primal and dual connected? Let $\boldsymbol{w}^*$ and $\{\lambda\}_j^*$ be the primal and dual solutions respectively, then

$$\begin{aligned}\max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right) &= \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j^*\}\right) \leq L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) \\ &\leq \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}^*, \{\lambda_j\}\right)\end{aligned}$$

Duality

We define the **dual problem** by swapping the min and max:

$$\max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

How are the primal and dual connected? Let $\boldsymbol{w}^*$ and $\{\lambda\}_j^*$ be the primal and dual solutions respectively, then

$$\begin{aligned}\max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right) &= \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j^*\}\right) \leq L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) \\ &\leq \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}^*, \{\lambda_j\}\right) = \min_{\boldsymbol{w}} \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}, \{\lambda_j\}\right)\end{aligned}$$

Duality

We define the **dual problem** by swapping the min and max:

$$\max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

How are the primal and dual connected? Let $\boldsymbol{w}^*$ and $\{\lambda\}_j^*$ be the primal and dual solutions respectively, then

$$\begin{aligned}\max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right) &= \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j^*\}\right) \leq L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) \\ &\leq \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}^*, \{\lambda_j\}\right) = \min_{\boldsymbol{w}} \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}, \{\lambda_j\}\right)\end{aligned}$$

This is called "**weak duality**".

Strong duality

When $F, h_1, \ldots, h_m$ are convex, under some mild conditions:

$$\min_{\boldsymbol{w}} \max_{\{\lambda_j\} \geq 0} L\left(\boldsymbol{w}, \{\lambda_j\}\right) = \max_{\{\lambda_j\} \geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

Strong duality

When $F, h_1, \ldots, h_m$ are convex, under some mild conditions:

$$\min_{\boldsymbol{w}} \max_{\{\lambda_j\} \geq 0} L\left(\boldsymbol{w}, \{\lambda_j\}\right) = \max_{\{\lambda_j\} \geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

This is called "**strong duality**".

Deriving the Karush-Kuhn-Tucker (KKT) conditions

Observe that if strong duality holds:

$$F(\boldsymbol{w}^*) = \min_{\boldsymbol{w}} \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}, \{\lambda_j\}\right) = \max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

Deriving the Karush-Kuhn-Tucker (KKT) conditions

Observe that if strong duality holds:

$$F(\boldsymbol{w}^*) = \min_{\boldsymbol{w}} \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}, \{\lambda_j\}\right) = \max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

$$= \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j^*\}\right)$$

Deriving the Karush-Kuhn-Tucker (KKT) conditions

Observe that if strong duality holds:

$$F(\boldsymbol{w}^*) = \min_{\boldsymbol{w}} \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}, \{\lambda_j\}\right) = \max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

$$= \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j^*\}\right) \leq L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right)$$

Deriving the Karush-Kuhn-Tucker (KKT) conditions

Observe that if strong duality holds:

$$F(\boldsymbol{w}^*) = \min_{\boldsymbol{w}} \max_{\{\lambda_j\}\geq 0} L(\boldsymbol{w}, \{\lambda_j\}) = \max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L(\boldsymbol{w}, \{\lambda_j\})$$

$$= \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j^*\}\right) \leq L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) = F(\boldsymbol{w}^*) + \sum_{j=1}^{\mathsf{J}} \lambda_j^* h_j(\boldsymbol{w}^*)$$

Deriving the Karush-Kuhn-Tucker (KKT) conditions

Observe that if strong duality holds:

$$F(\boldsymbol{w}^*) = \min_{\boldsymbol{w}} \max_{\{\lambda_j\} \geq 0} L(\boldsymbol{w}, \{\lambda_j\}) = \max_{\{\lambda_j\} \geq 0} \min_{\boldsymbol{w}} L(\boldsymbol{w}, \{\lambda_j\})$$

$$= \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j^*\}\right) \leq L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) = F(\boldsymbol{w}^*) + \sum_{j=1}^{\mathsf{J}} \lambda_j^* h_j(\boldsymbol{w}^*) \leq F(\boldsymbol{w}^*)$$

Deriving the Karush-Kuhn-Tucker (KKT) conditions

Observe that if strong duality holds:

$$F(\boldsymbol{w}^*) = \min_{\boldsymbol{w}} \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}, \{\lambda_j\}\right) = \max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

$$= \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j^*\}\right) \leq L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) = F(\boldsymbol{w}^*) + \sum_{j=1}^{\mathsf{J}} \lambda_j^* h_j(\boldsymbol{w}^*) \leq F(\boldsymbol{w}^*)$$

Implications:

- *all inequalities above have to be equalities!*

Deriving the Karush-Kuhn-Tucker (KKT) conditions

Observe that if strong duality holds:

$$F(\boldsymbol{w}^*) = \min_{\boldsymbol{w}} \max_{\{\lambda_j\}\geq 0} L(\boldsymbol{w}, \{\lambda_j\}) = \max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L(\boldsymbol{w}, \{\lambda_j\})$$

$$= \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j^*\}\right) \leq L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) = F(\boldsymbol{w}^*) + \sum_{j=1}^{\mathsf{J}} \lambda_j^* h_j(\boldsymbol{w}^*) \leq F(\boldsymbol{w}^*)$$

Implications:

- *all inequalities above have to be equalities!*
- last equality implies $\lambda_j^* h_j(\boldsymbol{w}^*) = 0$ for all $j \in [\mathsf{J}]$

Deriving the Karush-Kuhn-Tucker (KKT) conditions

Observe that if strong duality holds:

$$F(\boldsymbol{w}^*) = \min_{\boldsymbol{w}} \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}, \{\lambda_j\}\right) = \max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

$$= \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j^*\}\right) \leq L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) = F(\boldsymbol{w}^*) + \sum_{j=1}^{\mathsf{J}} \lambda_j^* h_j(\boldsymbol{w}^*) \leq F(\boldsymbol{w}^*)$$

Implications:

- *all inequalities above have to be equalities!*
- last equality implies $\lambda_j^* h_j(\boldsymbol{w}^*) = 0$ for all $j \in [\mathsf{J}]$
- equality $\min_{\boldsymbol{w}} L(\boldsymbol{w}, \{\lambda_j^*\}) = L(\boldsymbol{w}^*, \{\lambda_j^*\})$ implies $\boldsymbol{w}^*$ is a **minimizer** of $L(\boldsymbol{w}, \{\lambda_j^*\})$

Deriving the Karush-Kuhn-Tucker (KKT) conditions

Observe that if strong duality holds:

$$F(\boldsymbol{w}^*) = \min_{\boldsymbol{w}} \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}, \{\lambda_j\}\right) = \max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

$$= \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j^*\}\right) \leq L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) = F(\boldsymbol{w}^*) + \sum_{j=1}^{\mathsf{J}} \lambda_j^* h_j(\boldsymbol{w}^*) \leq F(\boldsymbol{w}^*)$$

Implications:

- *all inequalities above have to be equalities!*
- last equality implies $\lambda_j^* h_j(\boldsymbol{w}^*) = 0$ for all $j \in [\mathsf{J}]$
- equality $\min_{\boldsymbol{w}} L(\boldsymbol{w}, \{\lambda_j^*\}) = L(\boldsymbol{w}^*, \{\lambda_j^*\})$ implies $\boldsymbol{w}^*$ is a **minimizer** of $L(\boldsymbol{w}, \{\lambda_j^*\})$ and thus has **zero gradient**:

$$\nabla_{\boldsymbol{w}} L(\boldsymbol{w}^*, \{\lambda_j^*\}) = \nabla F(\boldsymbol{w}^*) + \sum_{j=1}^{\mathsf{J}} \lambda_j^* \nabla h_j(\boldsymbol{w}^*) = \boldsymbol{0}$$

The Karush-Kuhn-Tucker (KKT) conditions

If $\boldsymbol{w}^*$ and $\{\lambda_j^*\}$ are the primal and dual solution respectively, then:

The Karush-Kuhn-Tucker (KKT) conditions

If $\boldsymbol{w}^*$ and $\{\lambda_j^*\}$ are the primal and dual solution respectively, then:

Stationarity:

$$\nabla_{\boldsymbol{w}} L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) = \nabla F(\boldsymbol{w}^*) + \sum_{j=1}^{\mathsf{J}} \lambda_j^* \nabla h_j(\boldsymbol{w}^*) = \mathbf{0}$$

The Karush-Kuhn-Tucker (KKT) conditions

If $\boldsymbol{w}^*$ and $\{\lambda_j^*\}$ are the primal and dual solution respectively, then:

Stationarity:

$$\nabla_{\boldsymbol{w}} L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) = \nabla F(\boldsymbol{w}^*) + \sum_{j=1}^{\mathsf{J}} \lambda_j^* \nabla h_j(\boldsymbol{w}^*) = \boldsymbol{0}$$

Complementary slackness:

$$\lambda_j^* h_j(\boldsymbol{w}^*) = 0 \quad \text{for all } j \in [\mathsf{J}]$$

The Karush-Kuhn-Tucker (KKT) conditions

If $\boldsymbol{w}^*$ and $\{\lambda_j^*\}$ are the primal and dual solution respectively, then:

Stationarity:

$$\nabla_{\boldsymbol{w}} L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) = \nabla F(\boldsymbol{w}^*) + \sum_{j=1}^{\mathsf{J}} \lambda_j^* \nabla h_j(\boldsymbol{w}^*) = \mathbf{0}$$

Complementary slackness:

$$\lambda_j^* h_j(\boldsymbol{w}^*) = 0 \quad \text{for all } j \in [\mathsf{J}]$$

Feasibility:

$$h_j(\boldsymbol{w}^*) \leq 0 \quad \text{and} \quad \lambda_j^* \geq 0 \quad \text{for all } j \in [\mathsf{J}]$$

The Karush-Kuhn-Tucker (KKT) conditions

If $\boldsymbol{w}^*$ and $\{\lambda_j^*\}$ are the primal and dual solution respectively, then:

Stationarity:

$$\nabla_{\boldsymbol{w}} L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) = \nabla F(\boldsymbol{w}^*) + \sum_{j=1}^{\mathsf{J}} \lambda_j^* \nabla h_j(\boldsymbol{w}^*) = \mathbf{0}$$

Complementary slackness:

$$\lambda_j^* h_j(\boldsymbol{w}^*) = 0 \quad \text{for all } j \in [\mathsf{J}]$$

Feasibility:

$$h_j(\boldsymbol{w}^*) \le 0 \quad \text{and} \quad \lambda_j^* \ge 0 \quad \text{for all } j \in [\mathsf{J}]$$

These are *necessary conditions*.

The Karush-Kuhn-Tucker (KKT) conditions

If $\boldsymbol{w}^*$ and $\{\lambda_j^*\}$ are the primal and dual solution respectively, then:

Stationarity:

$$\nabla_{\boldsymbol{w}} L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) = \nabla F(\boldsymbol{w}^*) + \sum_{j=1}^{\mathsf{J}} \lambda_j^* \nabla h_j(\boldsymbol{w}^*) = \boldsymbol{0}$$

Complementary slackness:

$$\lambda_j^* h_j(\boldsymbol{w}^*) = 0 \quad \text{for all } j \in [\mathsf{J}]$$

Feasibility:

$$h_j(\boldsymbol{w}^*) \leq 0 \quad \text{and} \quad \lambda_j^* \geq 0 \quad \text{for all } j \in [\mathsf{J}]$$

These are *necessary conditions*. They are also *sufficient* when F is convex and $h_1, \ldots, h_{\mathsf{J}}$ are continuously differentiable convex functions.

Outline

Writing down the Lagrangian

Recall the primal formulation

$$\begin{aligned}\min_{\boldsymbol{w},b,\{\xi_n\}} \quad & C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \leq \xi_n, \quad \forall\ n \\ & \xi_n \geq 0, \quad \forall\ n\end{aligned}$$

Writing down the Lagrangian

Recall the primal formulation

$$\begin{aligned} \min_{\boldsymbol{w}, b, \{\xi_n\}} \quad & C \sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & 1 - y_n[\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b] \leq \xi_n, \quad \forall\ n \\ & \xi_n \geq 0, \quad \forall\ n \end{aligned}$$

Lagrangian is

$$\begin{aligned} L\left(\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}\right) = {} & C \sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_n \lambda_n \xi_n \\ & + \sum_n \alpha_n \left(1 - y_n(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b) - \xi_n\right) \end{aligned}$$

where $\alpha_1, \ldots, \alpha_{\mathsf{N}} \geq 0$ and $\lambda_1, \ldots, \lambda_{\mathsf{N}} \geq 0$ are Lagrangian multipliers.

Applying the stationarity condition

$$L = C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \left(1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) - \xi_n\right)$$

$\exists$ primal and dual variables $\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}$ s.t. $\nabla_{\boldsymbol{w},b,\{\xi_n\}} L = \mathbf{0}$,

Applying the stationarity condition

$$L = C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \left(1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) - \xi_n\right)$$

$\exists$ primal and dual variables $\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}$ s.t. $\nabla_{\boldsymbol{w}, b, \{\xi_n\}} L = \boldsymbol{0}$, which means

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n) = \boldsymbol{0}$$

Applying the stationarity condition

$$L = C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \left(1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) - \xi_n\right)$$

$\exists$ primal and dual variables $\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}$ s.t. $\nabla_{\boldsymbol{w},b,\{\xi_n\}} L = \mathbf{0}$, which means

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n) = \mathbf{0} \quad \Longrightarrow \quad \boldsymbol{w} = \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)$$

Applying the stationarity condition

$$L = C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \left(1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) - \xi_n\right)$$

$\exists$ primal and dual variables $\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}$ s.t. $\nabla_{\boldsymbol{w}, b, \{\xi_n\}} L = \mathbf{0}$, which means

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n) = \mathbf{0} \quad \Longrightarrow \quad \boldsymbol{w} = \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)$$

$$\frac{\partial L}{\partial b} = -\sum_n \alpha_n y_n = 0$$

Applying the stationarity condition

$$L = C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \left(1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) - \xi_n\right)$$

$\exists$ primal and dual variables $\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}$ s.t. $\nabla_{\boldsymbol{w}, b, \{\xi_n\}} L = \mathbf{0}$, which means

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n) = \mathbf{0} \quad \Longrightarrow \quad \boldsymbol{w} = \sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)$$

$$\frac{\partial L}{\partial b} = -\sum_n \alpha_n y_n = 0 \quad \text{and} \quad \frac{\partial L}{\partial \xi_n} = C - \lambda_n - \alpha_n = 0, \quad \forall\, n$$

Rewrite the Lagrangian in terms of dual variables

Replacing $\boldsymbol{w}$ by $\sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)$ in the Lagrangian gives

$$L = C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \left(1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) - \xi_n\right)$$

Rewrite the Lagrangian in terms of dual variables

Replacing $\boldsymbol{w}$ by $\sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)$ in the Lagrangian gives

$$L = C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \left(1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) - \xi_n\right)$$

$$= C\sum_n \xi_n + \frac{1}{2}\|\sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)\|_2^2 - \sum_n \lambda_n \xi_n +$$

$$\sum_n \alpha_n \left(1 - y_n\left(\left(\sum_m y_m \alpha_m \boldsymbol{\phi}(\boldsymbol{x}_m)\right)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b\right) - \xi_n\right)$$

Rewrite the Lagrangian in terms of dual variables

Replacing w by $\sum_n y_n \alpha_n \phi(x_n)$ in the Lagrangian gives

$$L = C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \left(1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) - \xi_n\right)$$

$$= C\sum_n \xi_n + \frac{1}{2}\|\sum_n y_n\alpha_n\boldsymbol{\phi}(\boldsymbol{x}_n)\|_2^2 - \sum_n \lambda_n\xi_n +$$

$$\sum_n \alpha_n \left(1 - y_n\left(\left(\sum_m y_m\alpha_m\boldsymbol{\phi}(\boldsymbol{x}_m)\right)^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b\right) - \xi_n\right)$$

$$= \sum_n \alpha_n + \frac{1}{2}\|\sum_n y_n\alpha_n\boldsymbol{\phi}(\boldsymbol{x}_n)\|_2^2 - \sum_{m,n}\alpha_n\alpha_m y_m y_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n)$$

($\sum_n \alpha_n y_n = 0$ and $C = \lambda_n + \alpha_n$)

Rewrite the Lagrangian in terms of dual variables

Replacing $\boldsymbol{w}$ by $\sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)$ in the Lagrangian gives

$$L = C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \left(1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) - \xi_n\right)$$

$$= C\sum_n \xi_n + \frac{1}{2}\|\sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)\|_2^2 - \sum_n \lambda_n \xi_n +$$

$$\sum_n \alpha_n \left(1 - y_n \left(\left(\sum_m y_m \alpha_m \boldsymbol{\phi}(\boldsymbol{x}_m)\right)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b\right) - \xi_n\right)$$

$$= \sum_n \alpha_n + \frac{1}{2}\|\sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)\|_2^2 - \sum_{m,n} \alpha_n \alpha_m y_m y_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n)$$

$$\left(\sum_n \alpha_n y_n = 0 \text{ and } C = \lambda_n + \alpha_n\right)$$

$$= \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} \alpha_n \alpha_m y_m y_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n)$$

The dual formulation

To find the dual solutions, it amounts to solving

$$
\begin{aligned}
\max_{\{\alpha_n\},\{\lambda_n\}} \quad & \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) \\
\text{s.t.} \quad & \sum_n \alpha_n y_n = 0 \\
& C - \lambda_n - \alpha_n = 0, \ \alpha_n \geq 0, \ \lambda_n \geq 0, \quad \forall \ n
\end{aligned}
$$

The dual formulation

To find the dual solutions, it amounts to solving

$$\max_{\{\alpha_n\},\{\lambda_n\}} \quad \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n)$$

$$\text{s.t.} \quad \sum_n \alpha_n y_n = 0$$

$$C - \lambda_n - \alpha_n = 0, \; \alpha_n \geq 0, \; \lambda_n \geq 0, \quad \forall\, n$$

Note the last three constraints can be written as $0 \leq \alpha_n \leq C$ for all n.

The dual formulation

To find the dual solutions, it amounts to solving

$$\begin{aligned} \max_{\{\alpha_n\},\{\lambda_n\}} \quad & \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) \\ \text{s.t.} \quad & \sum_n \alpha_n y_n = 0 \\ & C - \lambda_n - \alpha_n = 0,\ \alpha_n \geq 0,\ \lambda_n \geq 0, \quad \forall\ n \end{aligned}$$

Note the last three constraints can be written as $0 \leq \alpha_n \leq C$ for all n. So the final **dual formulation of SVM** is:

$$\begin{aligned} \max_{\{\alpha_n\}} \quad & \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) \\ \text{s.t.} \quad & \sum_n \alpha_n y_n = 0 \quad \text{and} \quad 0 \leq \alpha_n \leq C, \quad \forall\ n \end{aligned}$$

Kernelizing SVM

Now it is clear that with a **kernel function** k for the mapping ϕ, we can kernelize SVM as:

$$\max_{\{\alpha_n\}} \quad \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$

$$\text{s.t.} \quad \sum_n \alpha_n y_n = 0 \quad \text{and} \quad 0 \leq \alpha_n \leq C, \quad \forall\ n$$

Again, no need to compute $\phi(\boldsymbol{x})$.

Kernelizing SVM

Now it is clear that with a **kernel function** k for the mapping ϕ, we can kernelize SVM as:

$$\max_{\{\alpha_n\}} \quad \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$

$$\text{s.t.} \quad \sum_n \alpha_n y_n = 0 \quad \text{and} \quad 0 \le \alpha_n \le C, \quad \forall\ n$$

Again, no need to compute $\phi(\boldsymbol{x})$. It is a **quadratic program** and many efficient optimization algorithms exist.

Recover the primal solution

But how do we predict given the dual solution $\{\alpha_n^*\}$?

Recover the primal solution

But how do we predict given the dual solution $\{\alpha_n^*\}$? Need to figure out the primal solution $\boldsymbol{w}^*$ and b^*.

Recover the primal solution

But how do we predict given the dual solution $\{\alpha_n^*\}$? Need to figure out the primal solution $\boldsymbol{w}^*$ and b^*.

Based on previous observation,

$$\boldsymbol{w}^* = \sum_n \alpha_n^* y_n \boldsymbol{\phi}(\boldsymbol{x}_n)$$

Recover the primal solution

But how do we predict given the dual solution $\{\alpha_n^*\}$? Need to figure out the primal solution $\boldsymbol{w}^*$ and b^*.

Based on previous observation,

$$\boldsymbol{w}^* = \sum_n \alpha_n^* y_n \boldsymbol{\phi}(\boldsymbol{x}_n) = \sum_{n:\alpha_n>0} \alpha_n^* y_n \boldsymbol{\phi}(\boldsymbol{x}_n)$$

Recover the primal solution

But how do we predict given the dual solution $\{\alpha_n^*\}$? Need to figure out the primal solution $\boldsymbol{w}^*$ and b^*.

Based on previous observation,

$$\boldsymbol{w}^* = \sum_n \alpha_n^* y_n \boldsymbol{\phi}(\boldsymbol{x}_n) = \sum_{n:\alpha_n > 0} \alpha_n^* y_n \boldsymbol{\phi}(\boldsymbol{x}_n)$$

A point with $\alpha_n^* > 0$ is called a "**support vector**". Hence the name SVM.

Recover the primal solution

But how do we predict given the dual solution $\{\alpha_n^*\}$? Need to figure out the primal solution $\boldsymbol{w}^*$ and b^*.

Based on previous observation,

$$\boldsymbol{w}^* = \sum_n \alpha_n^* y_n \boldsymbol{\phi}(\boldsymbol{x}_n) = \sum_{n:\alpha_n>0} \alpha_n^* y_n \boldsymbol{\phi}(\boldsymbol{x}_n)$$

A point with $\alpha_n^* > 0$ is called a "**support vector**". Hence the name SVM.

To identify b, we need to apply complementary slackness.

Applying complementary slackness

For all n we should have

$$\lambda_n^* \xi_n^* = 0, \quad \alpha_n^* \left(1 - \xi_n^* - y_n(\boldsymbol{w}^{*\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b^*)\right) = 0$$

Applying complementary slackness

For all n we should have

$$\lambda_n^* \xi_n^* = 0, \quad \alpha_n^* \left(1 - \xi_n^* - y_n(\boldsymbol{w}^{*\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b^*)\right) = 0$$

For some support vector $\boldsymbol{\phi}(\boldsymbol{x}_n)$ if we have $0 < \alpha_n^* < C$, then

$$\lambda_n^* = C - \alpha_n^* > 0$$

Applying complementary slackness

For all n we should have

$$\lambda_n^* \xi_n^* = 0, \quad \alpha_n^* \left(1 - \xi_n^* - y_n({\boldsymbol{w}^*}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b^*)\right) = 0$$

For some support vector $\boldsymbol{\phi}(\boldsymbol{x}_n)$ if we have $0 < \alpha_n^* < C$, then

$$\lambda_n^* = C - \alpha_n^* > 0$$

With the first condition we know $\xi_n^* = 0$.

Applying complementary slackness

For all n we should have

$$\lambda_n^* \xi_n^* = 0, \quad \alpha_n^* \left(1 - \xi_n^* - y_n(\boldsymbol{w}^{*\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b^*)\right) = 0$$

For some support vector $\boldsymbol{\phi}(\boldsymbol{x}_n)$ if we have $0 < \alpha_n^* < C$, then

$$\lambda_n^* = C - \alpha_n^* > 0$$

With the first condition we know $\xi_n^* = 0$.

With the second condition we know $1 = y_n(\boldsymbol{w}^{*\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b^*)$

Applying complementary slackness

For all n we should have

$$\lambda_n^* \xi_n^* = 0, \quad \alpha_n^* \left(1 - \xi_n^* - y_n({\boldsymbol{w}^*}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b^*)\right) = 0$$

For some support vector $\boldsymbol{\phi}(\boldsymbol{x}_n)$ if we have $0 < \alpha_n^* < C$, then

$$\lambda_n^* = C - \alpha_n^* > 0$$

With the first condition we know $\xi_n^* = 0$.

With the second condition we know $1 = y_n({\boldsymbol{w}^*}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b^*)$ and thus

$$b^* = y_n - {\boldsymbol{w}^*}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n)$$

Applying complementary slackness

For all n we should have

$$\lambda_n^* \xi_n^* = 0, \quad \alpha_n^* \left(1 - \xi_n^* - y_n({\boldsymbol{w}^*}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b^*)\right) = 0$$

For some support vector $\boldsymbol{\phi}(\boldsymbol{x}_n)$ if we have $0 < \alpha_n^* < C$, then

$$\lambda_n^* = C - \alpha_n^* > 0$$

With the first condition we know $\xi_n^* = 0$.

With the second condition we know $1 = y_n({\boldsymbol{w}^*}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b^*)$ and thus

$$b^* = y_n - {\boldsymbol{w}^*}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) = y_n - \sum_m y_m \alpha_m^* k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$

Applying complementary slackness

For all n we should have

$$\lambda_n^* \xi_n^* = 0, \quad \alpha_n^* \left(1 - \xi_n^* - y_n(\boldsymbol{w}^{*\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b^*)\right) = 0$$

For some support vector $\boldsymbol{\phi}(\boldsymbol{x}_n)$ if we have $0 < \alpha_n^* < C$, then

$$\lambda_n^* = C - \alpha_n^* > 0$$

With the first condition we know $\xi_n^* = 0$.

With the second condition we know $1 = y_n(\boldsymbol{w}^{*\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b^*)$ and thus

$$b^* = y_n - \boldsymbol{w}^{*\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) = y_n - \sum_m y_m \alpha_m^* k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$

The prediction on a new point $\boldsymbol{x}$ is therefore

$$\mathrm{SGN}\left(\boldsymbol{w}^{*\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}) + b^*\right) = \mathrm{SGN}\left(\sum_m y_m \alpha_m^* k(\boldsymbol{x}_m, \boldsymbol{x}) + b^*\right)$$

Geometric interpretation of support vectors

A support vector satisfies $\alpha_n^* \neq 0$ and

$$1 - \xi_n^* - y_n(\boldsymbol{w}^{*\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b^*) = 0$$

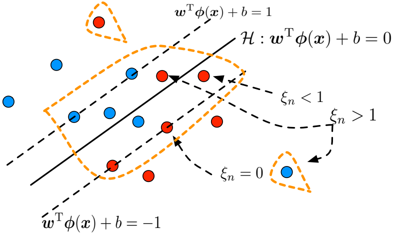
Geometric interpretation of support vectors

A support vector satisfies $\alpha_n^* \neq 0$ and

$$1 - \xi_n^* - y_n(\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b^*) = 0$$

When

- $\xi_n^* = 0$, $y_n(\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b^*) = 1$ and thus the point is $1/\|\boldsymbol{w}^*\|_2$ away from the hyperplane.



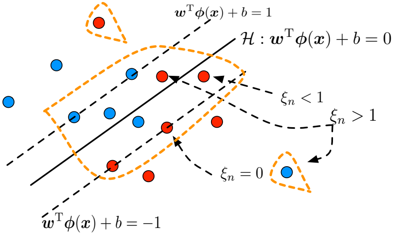
Geometric interpretation of support vectors

A support vector satisfies $\alpha_n^* \neq 0$ and

$$1 - \xi_n^* - y_n(\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b^*) = 0$$

When

- $\xi_n^* = 0$, $y_n(\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b^*) = 1$ and thus the point is $1/\|\boldsymbol{w}^*\|_2$ away from the hyperplane.
- $\xi_n^* < 1$, the point is classified correctly but does not satisfy the large margin constraint.



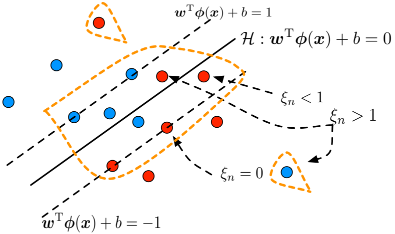
Geometric interpretation of support vectors

A support vector satisfies $\alpha_n^* \neq 0$ and

$$1 - \xi_n^* - y_n(\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b^*) = 0$$

When

- $\xi_n^* = 0$, $y_n(\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b^*) = 1$ and thus the point is $1/\|\boldsymbol{w}^*\|_2$ away from the hyperplane.
- $\xi_n^* < 1$, the point is classified correctly but does not satisfy the large margin constraint.
- $\xi_n^* > 1$, the point is misclassified.



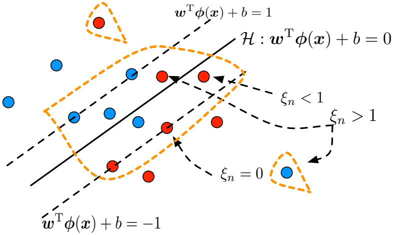
Geometric interpretation of support vectors

A support vector satisfies $\alpha_n^* \neq 0$ and

$$1 - \xi_n^* - y_n(\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b^*) = 0$$

When

- $\xi_n^* = 0$, $y_n(\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b^*) = 1$ and thus the point is $1/\|\boldsymbol{w}^*\|_2$ away from the hyperplane.
- $\xi_n^* < 1$, the point is classified correctly but does not satisfy the large margin constraint.
- $\xi_n^* > 1$, the point is misclassified.



Support vectors (circled with the orange line) are *the only points that matter!*

An example

One drawback of kernel method: **non-parametric**, need to keep all training points potentially

An example

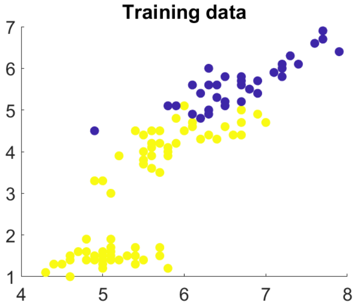
One drawback of kernel method: **non-parametric**, need to keep all training points potentially

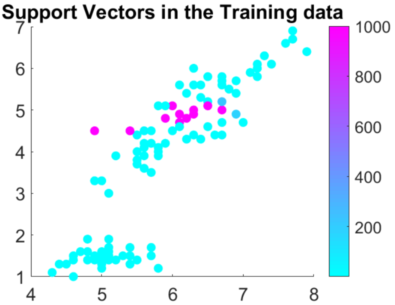
For SVM, very often $\#$support vectors $\ll N$

An example

One drawback of kernel method: **non-parametric**, need to keep all training points potentially

For SVM, very often #support vectors $\ll$ N





Summary

SVM: **max-margin linear classifier**

Summary

SVM: **max-margin linear classifier**

Primal (equivalent to minimizing L2 regularized hinge loss):

$$\begin{aligned} \min_{\boldsymbol{w}, b, \{\xi_n\}} \quad & C \sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & 1 - y_n(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b) \leq \xi_n, \quad \forall\, n \\ & \xi_n \geq 0, \quad \forall\, n \end{aligned}$$

Summary

SVM: **max-margin linear classifier**

Primal (equivalent to minimizing L2 regularized hinge loss):

$$\begin{aligned} \min_{\boldsymbol{w}, b, \{\xi_n\}} \quad & C \sum_n \xi_n + \frac{1}{2} \|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & 1 - y_n(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b) \leq \xi_n, \quad \forall\ n \\ & \xi_n \geq 0, \quad \forall\ n \end{aligned}$$

Dual (kernelizable, reveals what training points are support vectors):

$$\begin{aligned} \max_{\{\alpha_n\}} \quad & \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) \\ \text{s.t.} \quad & \sum_n \alpha_n y_n = 0 \quad \text{and} \quad 0 \leq \alpha_n \leq C, \quad \forall\ n \end{aligned}$$

Summary

Typical steps of applying Lagrangian duality

- start with a primal problem

Summary

Typical steps of applying Lagrangian duality

- start with a primal problem
- write down the Lagrangian (one dual variable per constraint)

Summary

Typical steps of applying Lagrangian duality

- start with a primal problem
- write down the Lagrangian (one dual variable per constraint)
- apply KKT conditions to find the connections between primal and dual solutions

Summary

Typical steps of applying Lagrangian duality

- start with a primal problem
- write down the Lagrangian (one dual variable per constraint)
- apply KKT conditions to find the connections between primal and dual solutions
- eliminate primal variables and arrive at the dual formulation

Summary

Typical steps of applying Lagrangian duality

- start with a primal problem
- write down the Lagrangian (one dual variable per constraint)
- apply KKT conditions to find the connections between primal and dual solutions
- eliminate primal variables and arrive at the dual formulation
- maximize the Lagrangian with respect to dual variables

Summary

Typical steps of applying Lagrangian duality

- start with a primal problem
- write down the Lagrangian (one dual variable per constraint)
- apply KKT conditions to find the connections between primal and dual solutions
- eliminate primal variables and arrive at the dual formulation
- maximize the Lagrangian with respect to dual variables
- recover the primal solutions from the dual solutions